笔试必备的计算机综合习题&答案--天天向上求职工作室独家整理复习笔记知识点

1.1 有 A、B、C、D 四个人，要在夜里过一座桥。

他们通过这座桥分别需要耗时 1、2、5、10 分钟，只有一支手电，并且同时最多只能两个人一起过桥。

请问，如何安排，能够在 17 分钟内这四个人都过桥？

答案：A & B -->2 mins

1 mins <-- A

C & D -->10 mins

2 mins <-- B

A & B --> 2 mins

一共 2 ＋ 1 ＋ 10 ＋ 2 ＋ 2 ＝ 17 mins

1.2 1-20 的两个数把和告诉 A，积告诉 B，A 说不知道是多少，

B 也说不知道，这时 A 说我知道了，B 接着说我也知道了，问这两个数是多少？

答案：2 和 3

1.3 爸爸，妈妈，妹妹，小强，至少两个人同一生肖的概率是多少?

1-12*11*10*9/12*12*12*12 = 1-55/96 = 41/96

1.4 某人去玩具店买小熊，单价 30 元. 付给玩具店老板 100 元玩具店老板没零钱，

去水果店换了 100 元零钱回来找给那人 70 元. 那人走后， 水果店老板找到玩具店老板说刚才的 100 元是假币，玩具店老板赔偿了水果店老板 100 元。

问：玩具店老板损失了多少钱?

答案： 70＋小熊的进价

2 请定义一个宏，比较两个数 a、b 的大小，不能使用大于、小于、if 语句

答案：

#define max(a，b) ((((long)((a)-(b)))&0x80000000)?b:a)

若 a > b ，则 a-b 的二进制最高位为 0，与上任何数还是 0，所以大数为 a；

否则，a-b 为负数，二进制最高位为 1，与上 0x80000000（最高位为 1 其他为 0）之后为 1，所以此时的大数为 b.

3 计算 a^b << 2

答案：运算符优先级：括号，下标，->和.(成员)最高：

单目的比双目的高；

算术双目的比其他双目的高；

位运算 高于 关系运算；

关系运算 高于 按位运算（与，或，异或）；

按位运算 高于 逻辑运算；

三目的只有一个 条件运算，低于逻辑运算；

赋值运算仅比 ， （顺序运算）高。

在此题中，位左移"<<" 优先级高于按位异或"^"，所以 b 先左移两位（相当于乘以 4），再与 a 异或。

例如： 当 a = 6； b = 4 时； 则 a^b<<2 = 22

4 如何输出源文件的标题和目前执行行的行数？
答案： printf("The file name: %d\n"， __FILE__);
printf("The current line No:%d\n"， __LINE__);
ANSI C 标准预定义宏:
_ _LINE_ _
_ _FILE_ _
_ _DATE_ _
_ _TIME_ _
_ _STDC_ _ 当要求程序严格遵循 ANSI C 标准时该标识符被赋值为 1
__cplusplus__ 当编写 C++程序时该标识符被定义

5 a[3][4]哪个不能表示 a[1][1]: *(&a[0][0]+5) *(*(a+1)+1) *(&a[1]+1) *(&a[0][0]+4)
答案: *(*(a+1)+1)
a 是数组的首地址，a+1 相当于&a[0][1]， *(a+1)=a[0][1]，则*(*(a+1)+1)= *(a[0][1] + 1) =/= a[1][1]

6 fun((exp1，exp2)，(exp3，exp4，exp5))有几个实参？
答案：两个。
形式参数：在声明和定义函数时，写在函数名后的括号中的参数。
实参是调用参数中的变量，行参是被调用函数中的变量。

7. 希尔 冒泡 快速 插入 哪个平均速度最快？
答案：快速排序
快速排序、归并排序和基数排序在不同情况下都是最快最有用的。

8. enum 的声明方式
答案：enum 枚举类型名 {
枚举常量 1，
枚举常量 2，
…
枚举常量 n
};
For example:
enum weekday { sunday， monday， tuesday， wednesday， thursday， friday， saturday};
enum weekday week_day;//week_day 就是一个枚举类型变量

9. 频繁的插入删除操作使用什么结构比较合适，链表还是数组？
答案：链表

10. *p=NULL *p=new char[100] sizeof(p)各为多少？
答案：都为 4。因为都是指针类型，所占存储空间必然为 4。

11. 顺序查找的平均时间

答案：(1+2+3+...+n)/n = (n+1)/2
12. for(i=0，sum=0; i<10; ++i，sum+=i);的运行结果
答案：sum = 55

13. 不能做 switch()的参数类型是：
答案：switch 的参数不能为浮点型。

14.不使用其他变量，交换两个整型 a，b 的值
答案：x = x+y; y = x-y; x = x-y

15. 写出 float x 与“零值”比较的 if 语句。
if(x>=0.000001 && x<=-0.000001)（x 不为 0 的比较）
float: 6 位精度
double: 16 位精度
16.两个数相乘，小数点后位数没有限制，请写一个高精度算法
****
数据库
****
1. 有个表 tableQQ，有整型的 ID 项和字符类型的 Nickname 项，这两个项都不允许为空
(1)写出建立该表的 SQL 语句
(2)找出 Nickname 为 QQ 的用户，按 ID 降序排列的 SQL 语句
(3)写出删除 ID 为 1234 用户记录的 SQL 语句
(4)写出添加 ID 为 5555，Nickname 为'1234'的 SQL 语句
答案：
(1) CREATE TABLE tableQQ
(
ID NUMBER(12) NOT NULL，
Nickname Varchar2(30) NOT NULL
);
(2) select * from tableQQ where Nickname = 'QQ' order by ID desc;
(3) delete from tableQQ where >
(4) insert into tableQQ values(5555，'1234');
//删除表
(5)drop table tableQQ;
2. 有关系 s(sno，sname) c(cno，cname) sc(sno，cno，grade)
1 问上课程 "db"的学生
2 成绩最高的学生号
3 每科大于 90 分的人数
答案：
(1)select a.sno， a.cno， b.cno， b.cname from sc a， c b where a.cno = b.cn
o and b.cname = 'db';
(2)select sno， max(grade)from sc group by sno;
(3)select cno， count(sno) from sc where grade > 90 group by cno;
****

操作系统 网络

1. 描述实时系统的基本特性

答案：在特定时间内完成特定的任务，实时性与可靠性。

2. Internet 采用哪种网络协议？该协议的主要层次结构？

答案：TCP/IP 协议。应用层、传输层、网络层、数据链路层和物理层。

3. Internet 物理地址和 IP 地址转换采用什么协议？

答案：地址解析协议 ARP address resolution protocol

4. IP 地址的编码分为哪俩部分？

答案：网络号和主机号。不过是要和“子网掩码”按位与上之后才能区分哪些是网络位
哪些是主机位。

二分查找

快速排序

双向链表的删除结点

有 12 个小球，外形相同，其中一个小球的质量与其他 11 个不同
给一个天平，问如何用 3 次把这个小球找出来
并且求出这个小球是比其他的轻还是重
解答:

哈哈，据说这是微软前几年的一个面试题。很经典滴啊！三次一定能求出来，而且能确定是重还是轻。

数据结构的知识还没怎么学透，不过这个题我到是自己研究过，可以分析下。

将 12 个球分别编号为 a1，a2，a3.....a10，a11，a12.

第一步：将 12 球分开 3 拨，每拨 4 个，a1~a4 第一拨，记为 b1， a5~a6 第 2 拨，记为 b2，其余第 3 拨，记为 b3；

第二步：将 b1 和 b2 放到天平两盘上，记左盘为 c1，右为 c2；这时候分两中情况：

1.c1 和 c2 平衡，此时可以确定从 a1 到 a8 都是常球；然后把 c2 拿空，并从 c1 上拿下 a4，从 a9 到 a12 四球里随便取三球，假设为 a9 到 a11，放到 c2 上。此时 c1 上是 a1 到 a3，c2 上是 a9 到 a11。从这里又分三种情况：

A：天平平衡，很简单，说明没有放上去的 a12 就是异球，而到此步一共称了两次，所以将 a12 随便跟 11 个常球再称一次，也就是第三次，马上就可以确定 a12 是重还是轻；

B：若 c1 上升，则这次称说明异球为 a9 到 a11 三球中的一个，而且是比常球重。取下 c1 所有的球，并将 a8 放到 c1 上，将 a9 取下，比较 a8 和 a11（第三次称），如果平衡则说明从 c2 上取下的 a9 是偏重异球，如果不平衡，则偏向哪盘则哪盘里放的就是偏重异球；

C：若 c1 下降，说明 a9 到 a11 里有一个是偏轻异球。次种情况和 B 类似，所以接下来的步骤照搬 B 就是；

2.c1 和 c2 不平衡，这时候又分两种情况，c1 上升和 c1 下降，但是不管哪种情况都能说明 a9 到 a12 是常球。这步是解题的关键。也是这个题最妙的地方。

A：c1 上升，此时不能判断异球在哪盘也不能判断是轻还是重。取下 c1 中的 a2 到 a4 三球放一边，将 c2 中的 a5 和 a6 放到 c1 上，然后将常球 a9 放到 c2 上。至此，c1 上是 a1，a5 和 a6，c2 上是 a7，a8 和 a9。此时又分三中情况：

1）如果平衡，说明天平上所有的球都是常球，异球在从 c1 上取下 a2 到 a4 中。而且可以断定异球轻重。因为 a5 到 a8 都是常球，而第 2 次称的时候 c1 是上升的，所以 a2 到 a4 里必然有一个轻球。那么第三次称

就用来从 a2 到 a4 中找到轻球。这很简单，随便拿两球放到 c1 和 c2，平衡则剩余的为要找球，不平衡则哪边低则哪个为要找球；

2）c1 仍然保持上升，则说明要么 a1 是要找的轻球，要么 a7 和 a8 两球中有一个是重球（这步懂吧？好好想想，很简单的。因为 a9 是常球，而取下的 a2 到 a4 肯定也是常球，还可以推出换盘放置的 a5 和 a6 也是常球。所以要么 a1 轻，要么 a7 或 a8 重）。至此，还剩一次称的机会。只需把 a7 和 a8 放上两盘，平衡则说明 a1 是要找的偏轻异球，如果不平衡，则哪边高说明哪个是偏重异球；

3）如果换球称第 2 次后天平平衡打破，并且 c1 降低了，这说明异球肯定在换过来的 a5 和 a6 两求中，并且异球偏重，否则天平要么平衡要么保持 c1 上升。确定要找球是偏重之后，将 a5 和 a6 放到两盘上称第 3 次根据哪边高可以判定 a5 和 a6 哪个是重球；

B：第 1 次称后 c1 是下降的，此时可以将 c1 看成 c2，其实以后的步骤都同 A，所以就不必要再重复叙述了。至此，不管情况如何，用且只用三次就能称出 12 个外观手感一模一样的小球中有质量不同于其他 11 球的偏常的球。而且在称的过程中可以判定其是偏轻还是偏重。

给一个奇数阶 N 幻方，填入数字 1，2，3...N*N，使得横竖斜方向上的和都相同

答案:

```
#include<iostream>
#include<iomanip>
#include<cmath>
usingnamespace std;
int main()
{
 int n;
 cin>>n;
 int i;
 int **Matr=newint*[n];//动态分配二维数组
 for(i=0;i<n;++i)
        Matr[ i ]=newint[n];//动态分配二维数组
 //j=n/2 代表首行中间数作为起点，即 1 所在位置
 int j=n/2，num=1;//初始值
 i=0;
 while(num!=n*n+1)
 {
//往右上角延升，若超出则用%转移到左下角
        Matr[(i%n+n)%n][(j%n+n)%n]=num;
     //斜行的长度和 n 是相等的，超出则转至下一斜行
     if(num%n==0)
            i++;
   else
 &nb
sp;
          {
i--;
            j++;
       }
```

```
        num++;
 }
 for(i=0;i<n;i++)
 {
        for(j=0;j<n;++j)
            cout<<setw((int)log10(n*n)+4)<<Matr[ i][ j ];//格式控制
        cout<<endl<<endl;//格式控制
 }
for(i=0;i<n;++i)
        delete [ ]Matr[ i ];
return1;
}
```

腾讯的一道面试题:(与百度相似，可惜昨天百度死在这方面了)////

在一个文件中有 10G 个整数，乱序排列，要求找出中位数。内存限制为 2G。只写出思路即可。

答案:

1，把整数分成 256M 段，每段可以用 64 位整数保存该段数据个数，256M*8 = 2G 内存，先清 0

2，读 10G 整数，把整数映射到 256M 段中，增加相应段的记数

3，扫描 256M 段的记数，找到中位数的段和中位数的段前面所有段的记数，可以把其他段的内存释放

4，因中位数段的可能整数取值已经比较小（如果是 32bit 整数，当然如果是 64bit 整数的话，可以再次分段），对每个整数做一个记数，再读一次 10G 整数，只读取中位数段对应的整数，并设置记数。

5，对新的记数扫描一次，即可找到中位数。

如果是 32bit 整数，读 10G 整数 2 次，扫描 256M 记数一次，后一次记数因数量很小，可以忽略不记

(设是 32bit 整数，按无符号整数处理整数分成 256M 段？ 整数范围是 0 - 2^32 - 1 一共有 4G 种取值，4G/256M = 16，每 16 个数算一段 0-15 是 1 段，16-31 是一段，... 整数映射到 256M 段中？ 如果整数是 0-15，则增加第一段记数，如果整数是 16-31，则增加第二段记数，... 其实可以不用分 256M 段，可以分的段数少一写，这样在扫描记数段时会快一些，还能节省一些内存)

腾讯题二:

一个文件中有 40 亿个整数，每个整数为四个字节，内存为 1GB，写出一个算法：求出这个文件里的整数里不包含的一个整数

答:

方法一: 4 个字节表示的整数，总共只有 2^32 约等于 4G 个可能。

为了简单起见，可以假设都是无符号整数。

分配 500MB 内存，每一 bit 代表一个整数，刚好可以表示完 4 个字节的整数，初始值为 0。基本思想每读入一个数，就把它对应的 bit 位置为 1 ，处理完 40G 个数后，对 500M 的内存遍历，找出一个 bit 为 0 的位，输出对应的整数就是未出现的。

算法流程：

1）分配 5 0 0 M B 内存 buf，初始化为 0

2）unsigned int x=0x1;

```
for each int j in file
buf=buf |x < <j;
end
```

(3) for(unsigned int i=0; i <= 0xffffffff; i++)
if (!(buf & x < <i))
{
output(i);
break;
}
以上只是针对无符号的，有符号的整数可以依此类推。

方法二:
文件可以分段读啊，这个是 O（2n）算法，应该是很快的了，而且空间也允许的。
不过还可以构造更快的方法的，更快的方法主要是针对定位输出的整数优化算法。
思路大概是这样的，把值空间等分成若干个值段，比如值为无符号数，则
00000000H-00000FFFH
00001000H-00001FFFH
.....
0000F000H-0000FFFFH
.....
FFFFF000H-FFFFFFFFH
这样可以订立一个规则，在一个值段范围内的数第一次出现时，对应值段指示值 Xn=Xn+1，如果该值段的所有整数都出现过，则 Xn=1000H，这样后面输出定位时就可以直接跳过这个值段了，因为题目仅仅要求输出一个，这样可以大大减少后面对标志数值的遍历步骤。
理论上值段的划分有一定的算法可以快速的实现，比如利用位运算直接定位值段对应值进行计算。
腾讯面试题: 有 1 到 10w 这 10w 个数，去除 2 个并打乱次序，如何找出那两个数。（不准用位图!!）
位图解决:
位图的方法如下
假设待处理数组为 A[10w-2]
定义一个数组 B[10w]，这里假设 B 中每个元素占用 1 比特，并初始化为全 0
for(i=0;i <10w-2;i++)
{
B[A[i]]=1
}
那么 B 中不为零的元素即为缺少的数据
这种方法的效率非常高，是计算机中最常用的算法之一
其它方法:
求和以及平方和可以得到结果，不过可能求平方和运算量比较大（用 64 位 int 不会溢出）

腾讯面试题:
腾讯服务器每秒有 2w 个 QQ 号同时上线，找出 5min 内重新登入的 qq 号并打印出来。

解答: 第二题如果空间足够大，可以定义一个大的数组
a[qq 号]，初始为零，然后这个 qq 号登陆了就 a[qq 号]++
最后统计大于等于 2 的 QQ 号
这个用空间来代替时间

第二个题目，有不成熟的想法。
2w x 300s
所以用 6，000，000 个桶。删除超时的算法后面说，所以平均桶的大小是 1 。
假设 qq 号码一共有 10^10 个，所以每个桶装的 q 号码是 10^10 / (6 * 10^6) 个，这个是插入时候的最坏效率（插入同一个桶的时候是顺序查找插入位置的）。
qq 的节点结构和上面大家讨论的基本一样，增加一个指针指向输出列表，后面说。

```
struct QQstruct {
   num_type     qqnum;
   timestamp    last_logon_time;
   QQstruct     *pre;
   QQstruct     *next;
   OutPutList *out;       // 用于 free 节点的时候，顺便更新一下输出列表。
}
```

另外增加两个指针列表。
第一个大小 300 的循环链表，自带一个指向 QQStruct 的域，循环存 300 秒内的 qq 指针。时间一过就 free 掉， 所以保证所有桶占用的空间在 2w X 300 以内。
第二个是 输出列表， 就是存放题目需要输出的节点。
如果登陆的用户，5 分钟内完全没有重复的话，每秒 free 掉 2w 个节点。
不过在 free 的时候，要判断一下时间是不是真的超时，因为把节点入桶的时候，遇到重复的，会更新一下最后登陆的时间。当然啦，这个时候，要把这个 qq 号码放到需要输出的列表里面